OKTECH
Smart thinking

Runtime analysis of Java applications

OKTECH Profiler

István Soós
istvan.soos@oktech.hu

Agenda

OKTECH Smart thinking

At the end of the presentation, you will be aware of ...

- ... the importance of application profiling.
- ... various performance measurement techniques.
- ... the impact of profiling and accepting the limits.

- ... the actual status of OKTECH Profiler.
- ... planned features and developments.

Measuring... why?

OKTECH
Smart thinking

- "Memory is cheap..."
- "A little bigger machine will manage the load"...
- "Hosting is cheap..."
- "Just a new machine in the cloud..."

Is it really that cheap?

- There are direct costs (or savings) on the performance.
- If there is competition, someone will benchmark to it.
- The client's time is always expensive, don't make him wait.

Measuring... what?

OKTECH Smart thinking

You can always ask questions about the application:

- How much percentage does the ... processing take?
- Why is this page so slow for the users?
- Do we have an infrastructural bottleneck here?
- Are our synchronization routines perfect?
- Where is the query that involves the most post-processing while reading the results from the database?
- We do have a cache, is it used correctly?
- ...
- What is happening ... just right now?

The Good, the Bad and the...

The Good, the Bad and the...

OKTECH Smart thinking

The Good programmer:

- does produce code with no errors
- does produce code with no bottlenecks
- does not exists

The Bad programmer:

- does produce code that has bad synchronization points
- does produce code that consumes a lot of resources
- does not care about these

The Profiler-aware programmer:

- runs a profiler on the code
- checks the bottlenecks and the most time-consuming parts
- fixes the issues if possible

Profiling methods

OKTECH
Smart thinking

Instrumentation

- Manual (coding, coding and coding...)
- Compile-time (pre-binary)
- Binary translation (post-compile)
- Runtime instrumentation (pre-run)
- Runtime injection (on-the-fly)

Sampling

- Execution trace (where are we now?)
- Monitor values (memory, cpu times...)

Hypervisor

- "Virtualization" or "Simulator" (step the clock)

Instrumentation

OKTECH Smart thinking

```
protected void someMethod() {
    long start = System.nanoTime();
    // ...
    long end = System.nanoTime();
    Trace.report("someMethod()", (end-start) );
}

protected void otherMethod() {
   Trace.reportStart("otherMethod");
   try {
      // ...
   } finally {
      Trace.reportEnd("otherMethod");
   }
}
```

Systematic error of instrumentation

```
public void a() { b(); }
public void b() { c(); }
public void c() { for (int i =0; i<100; i++) d(i); }
...
```

The systematic error:

- measurement times are added to the executions
- on multiple level, this accumulates
- there is no way to eliminate all measurement time
- will be repeatedly the same on each measurements
- distorts on lower timing values impact the overall result more

Sampling

1. Do not specify what to measure, just give me one actual state
   - StackTraceElement[] Thread.getStackTrace()
2. Repeat this couple of times.
3. Explore the depth of mathematical statistics :)

- Adjustable, typically smaller overhead, but

- Natural uncertainty
- Different kind of statistics

Sampling: the stack trace

OKTECH Smart thinking

```
com.sun.jndi.rmi.registry.RegistryContext.lookup(RegistryContext.java:101)
com.sun.jndi.toolkit.url.GenericURLContext.lookup(GenericURLContext.java:185)

javax.naming.InitialContext.lookup(InitialContext.java:392)

javax.management.remote.rmi.RMIConnector.findRMIServerJNDI(RMIConnector.java:1886)
javax.management.remote.rmi.RMIConnector.findRMIServer(RMIConnector.java:1856)
javax.management.remote.rmi.RMIConnector.connect(RMIConnector.java:257)
javax.management.remote.rmi.RMIConnector.connect(RMIConnector.java:338)
javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:248)

hu.oktech.profiler.runtime.remote.RemoteJmxRuntime.start(RemoteJmxRuntime.java:65)
hu.oktech.profiler.runtime.remote.RemoteJmxProfiler.main(RemoteJmxProfiler.java:42)
```

Further observer effects

JVM effects:

- Running anything attached to a JVM will affect GC times, threads, IO stats, sometimes synchronization
- Instrumentation will make HotSpot different (less optimized)

CPU effects:

- Modern CPUs do advanced caching and parallel processing of instructions
- Sometimes low-level instructions are not processed in the designated order

OKTECH Profiler summary

OKTECH Smart thinking

- http://code.google.com/p/oktech-profiler/

- Local Java-agent, remote JMX connections
- Sampling profiler
- Instrumentation profiler (1.1+)
- Simple tree statistics
  - XML output (1.1+)

Planned features:

- Monitoring, alerting
- More statistics, more output formats
- More precise instrumentation and sampling

- Commercial support: http://oktech.hu/

Demo: remote JMX connection

OKTECH Smart thinking

```
java -jar hu.oktech.profiler-java-runtime-1.1-all.jar \
  remote.jmx.url=service:jmx:rmi:///jndi/rmi://localhost:8686/jmxrmi \
  remote.jmx.user=admin \
  remote.jmx.password=adminadmin
```

```
java -jar hu.oktech.profiler-java-runtime-1.1-all.jar \
  prop=remote.properties
```

Demo: local Java agent

```
java \
  -javaagent:hu.oktech.profiler-java-runtime-1.1-all.jar=prop=local.
properties \
  -cp ...  some.Main arg1 arg2

instrument=my.company.*!,org.apache.*,-org.apache.catalina.*

#instrument-methods=
```

Demo: checking the reports

```
java -jar hu.oktech.profiler-report-1.1-all.jar \
  input=tmp/profiler/2009-09-16-19-00-00.dump \
```

```
java -jar hu.oktech.profiler-report-1.1-all.jar \
  prop=report.properties
```

With OKTECH Profiler...

... we can see and measure:

- Method-level performance characteristics (both)
- Possible locking and synchronization bottlenecks (sampling)
- Mostly acceptable method timings (instr.)

... we cannot see and measure:

- Parameter-level timings (e.g. per-parameter times)
- Large memory consumption areas

Unfortunately it doesn't answer the management's questions directly, but we will work on that one too :)

Q & A

OKTECH Smart thinking

At the end of the presentation, you are aware of ...

- ... the importance of application profiling.
- ... various performance measurement techniques.
- ... the impact of profiling and accepting the limits.

- ... the actual status of OKTECH Profiler.
- ... planned features and developments.